

**Fundamentos de Algoritmia**  
**Grados en Ingeniería Informática. Grupos C Y F**

Examen Convocatoria Extraordinaria, 5 de Julio de 2021.

**Nombre:** \_\_\_\_\_ **Grupo:** \_\_\_\_\_

**Laboratorio:** \_\_\_\_\_ **Puesto:** \_\_\_\_\_ **Usuario de DOMjudge:** \_\_\_\_\_

## **Normas de realización del examen**

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y entregarlas en el juez automático accesible en la dirección <http://exacrc.domjudge/team>.
2. Escribe comentarios que expliquen tu solución, justifiquen por qué se ha hecho así y ayuden a entenderla. Calcula la complejidad de todas las funciones que implementes.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribe tu **nombre y apellidos** en un comentario en la primera línea de cada fichero que subas al juez.
5. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta **exclusivamente** el último envío que hayas realizado de cada ejercicio.

1. (4 puntos) Dado un vector  $v$  de  $n \geq 0$  enteros positivos y dos naturales  $k, l \geq 0$  tales que  $k \leq n$  y las  $k$  primeras posiciones de  $v$  contienen pares, se desea contar cuántos segmentos de longitud  $k$  cumplen que tienen menos ( $\leq$ ) de  $l$  posiciones conteniendo números impares.
- (0.25 puntos) Define un predicado  $noMas(v, i, j, l)$  que devuelva cierto si y solo si la cantidad de números del vector  $v$  contenidos entre las posiciones  $i$  (incluida) y  $j$  (excluida) que son impares (supuesto  $0 \leq i \leq j \leq v.size()$ ) es menor o igual que  $l$ .
  - (0.5 puntos) Utilizando el predicado  $noMas$ , especifica una función que dados  $v, n, k$  y  $l$  tales que  $k, l \geq 0, k \leq n$  y las  $k$  primeras posiciones de  $v$  contienen pares, devuelva el número de segmentos de longitud  $k$  que tienen menos ( $\leq$ ) de  $l$  posiciones conteniendo números impares. Si  $k = 0$  la función debe devolver  $n + 1$ .
  - (2.5 puntos) Diseña e implementa un algoritmo iterativo eficiente que resuelva el problema propuesto.
  - (0.5 puntos) Escribe el invariante del bucle que permite demostrar la corrección del mismo y proporciona una función de cota.
  - (0.25 puntos) Indica el coste asintótico del algoritmo en el caso peor y justifica adecuadamente tu respuesta.

## Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el valor del número de elementos  $n$ , el valor de  $k$  y  $l$  y a continuación los elementos de la secuencia.

## Salida

Por cada caso de prueba el programa escribirá una línea con el número de segmentos solicitado en el enunciado.

## Entrada de ejemplo

```
5
8 4 2
2 6 4 8 1 10 3 2
8 4 0
2 6 4 8 9 3 1 7
4 2 5
2 4 1 3
3 0 2
1 3 5
3 1 1
2 3 6
```

## Salida de ejemplo

```
5
1
3
4
3
```

**2.(2.5 puntos)** Un dgito de un nmero natural  $n$  se dice que es *respetuoso* si es mayor o igual que todos los dgitos de  $n$  que son ms significativos que l. El dgito ms significativo es por tanto siempre respetuoso. Por ejemplo, en  $n = 1234$  todos los dgitos son respetuosos, mientras que en 3159 solamente el 1 no lo es. Se desea contar cuntos dgitos de  $n$  son respetuosos.

Se pide:

1. (1.75 puntos) Escribe un algoritmo recursivo eficiente que permita resolver el problema para un nmero  $n$  dado. No est permitido almacenar en un vector auxiliar los dgitos del nmero.
2. (0.75 punto) Escribe la recurrencia que corresponde al coste de la funcin recursiva utilizando el nmero de dgitos de  $n$  como tamao del problema. Indica tambin a qu orden de complejidad asinttica pertenece dicho coste.

### Entrada

La entrada comienza con una lnea que contiene el nmero de casos de prueba. Cada caso de prueba contendr el nmero  $n$ .

### Salida

Por cada caso de prueba el programa escribir el nmero de dgitos respetuosos de ese caso.

### Entrada de ejemplo

```
5
1234
0
32
303
4329
```

### Salida de ejemplo

```
4
1
1
2
2
```

3. (3.5 puntos) Dado un número entero  $M$  y un vector de  $n \geq 0$  números naturales  $v$ , diseñar un algoritmo de vuelta atrás que determine si existe una forma de insertar **entre** los  $n$  números del vector (tal como están colocados en el vector) operaciones de suma y resta de forma que se obtenga el número  $M$  como resultado final. Si, por ejemplo,  $M = 9$  y  $v = [1, 3, 2, 1, 4, 3, 2, 1]$ , la respuesta deberá ser sí ya que  $1 + 3 + 2 - 1 + 4 + 3 - 2 - 1 = 9$ , mientras que si  $v = [2, 1, 3]$  y  $M = -1$ , la respuesta deberá ser no, ya que  $2 + 1 + 3 = 6$ ,  $2 + 1 - 3 = 0$ ,  $2 - 1 + 3 = 4$  y  $2 - 1 - 3 = -2$ . Implementa un algoritmo de vuelta atrás que resuelva el problema. El algoritmo debe detenerse en cuanto encuentre una forma de obtener  $M$ . Se valorarán las podas que permitan reducir el número de nodos explorados.

### Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá el número entero  $M$ , el número  $n$  de números naturales y la secuencia de dichos números.

### Salida

Por cada caso de prueba el programa escribirá en una línea NO si no se puede obtener  $M$ , y en caso contrario escribirá SI.

### Entrada de ejemplo

```
5
9 8
1 3 2 1 4 3 2 1
-1 3
2 1 3
4 2
1 5
0 0
0 1
3
```

### Salida de ejemplo

```
SI
NO
NO
SI
NO
```